IITK MINI MIPS

Assignment-8

Group Number: 27

Students: Mohit Parihar(230330), Ch.V.Sai Koushik(230312)

PDS1: Decide the Registers and Their Usage Protocol

We have decided to use the general convention of the MIPS
32 registers for the General Purpose registers
32 registers for the Floating Point
We have stored them in an array of 64 elements, where the first 32 are GPRs and the next 32 are FPRs

GPR:

$zero: Constant zero. (1)

$at: Reserved for the assembler. (2)

$v0, $v1: For function return values.(4)

$a0–$a3: For function arguments.(8)

$t0–$t7 / $s0–$s7: For temporary and saved variables respectively.(16/24)

$k0-$k1: Reserved by operating system (26)

$gp: Global pointer(27)

$sp: Stack pointer(28)

$fp: Frame pointer(29)

$ra: Return address(30)

$t8-$t9: More temporaries, not saved (more like Hi and Lo register)(32)

FPR: We have defined 32 registers for floating-point instructions

PDS2: Decide upon the size for instruction and data memory.

We have allocated 1024 memory locations, each 32 bits wide, for both instruction and data memory.

This results in:

- Instruction Memory Size: 1024 × 32 bits = 4 KB
- Data Memory Size: 1024 × 32 bits = 4 KB

This size is suitable for small- to medium-scale programs and test benches. It allows storage of up to 1024 32-bit instructions or data values.

PDS3: Design the instruction layout for R-, I-, and J-type instructions and their respective encoding methodologies.

We have again opted to use the general convention of the MIPS

R-type layout:

opcode	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

I-type layout:

opcode	rs	rt	immediate
6 bits	5 bits	5 bits	16 bits

J-type layout:

opcode	address
6 bits	26 bits